

# iOS 7 Tech Talks 2013



San Francisco



New York



Tokyo



Shanghai



Berlin



London

These are confidential sessions—please refrain from streaming, blogging, or taking pictures



# Architecting Modern iOS Apps

## Part 2

Dave DeLong  
App Frameworks Evangelist

These are confidential sessions—please refrain from streaming, blogging, or taking pictures

# Overview

# Overview





# Overview



Building for *A7*





APL0598339S0



GKBC40H11213

K3PE7E700FXG

3SC009151222



64-bit

# 64-bit Benefits

- 2x integer registers
- 2x floating point registers
- More efficient assembly code
- Objective-C Runtime optimizations
- Faster performance
- Universal code base
- Build for the future



# Your App Bundle

# Your App Bundle





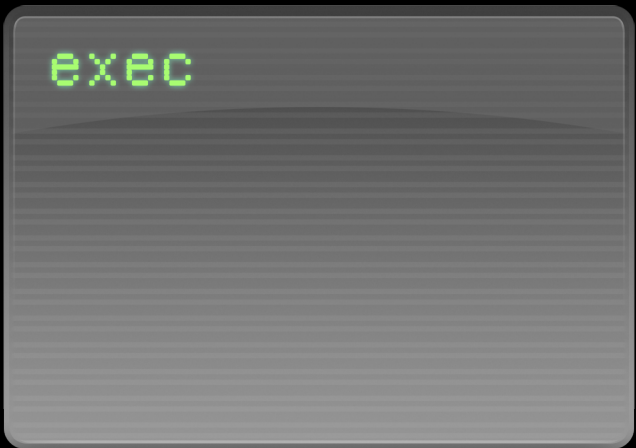
# Your App Bundle



# Your App Bundle



# Your App Bundle

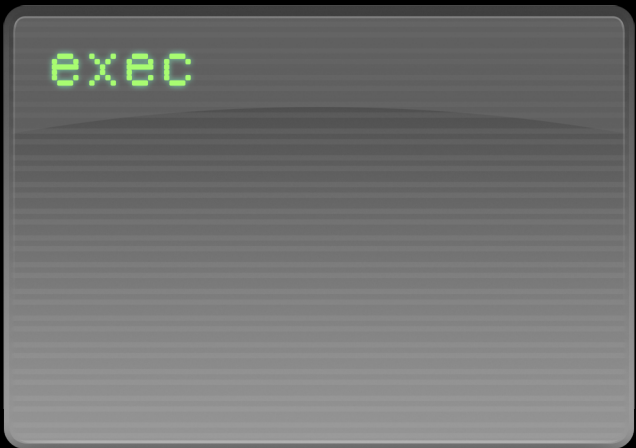


ApplicationGroup	Mine
CFBundleDevelopmentRegion	English
CFBundleExecutable	MyApp
CFBundleIconFile	MyAppIcon.icns
CFBundleIdentifier	com.me.myApp
CFBundleName	My Application
CFBundleGetInfoString	Copyright 2009, Me
CFBundleShortVersionString	1.0
CFBundleVersion	1.0
NSMainNibFile	MainMenu
NSPrincipalClass	MyClass

PLIST

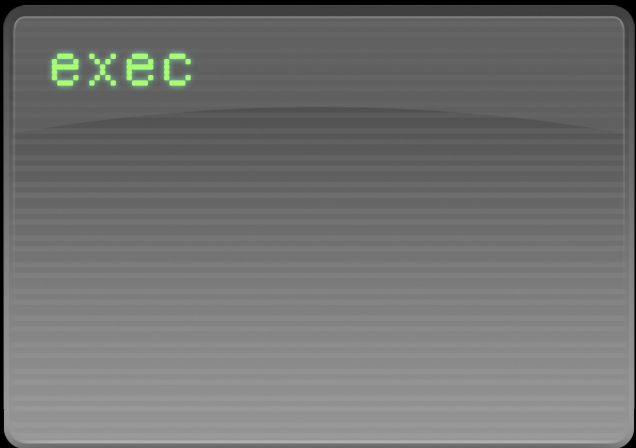


# Your App Bundle



# Your App Bundle

•Including 64-bit



ApplicationGroup	Mine
CFBundleDevelopmentRegion	English
CFBundleExecutable	MyApp
CFBundleIconFile	MyAppIcon.icns
CFBundleIdentifier	com.me.myApp
CFBundleName	My Application
CFBundleGetInfoString	Copyright 2009, Me
CFBundleShortVersionString	1.0
CFBundleVersion	1.0
NSMainNibFile	MainMenu
NSPrincipalClass	MyClass

PLIST

exec



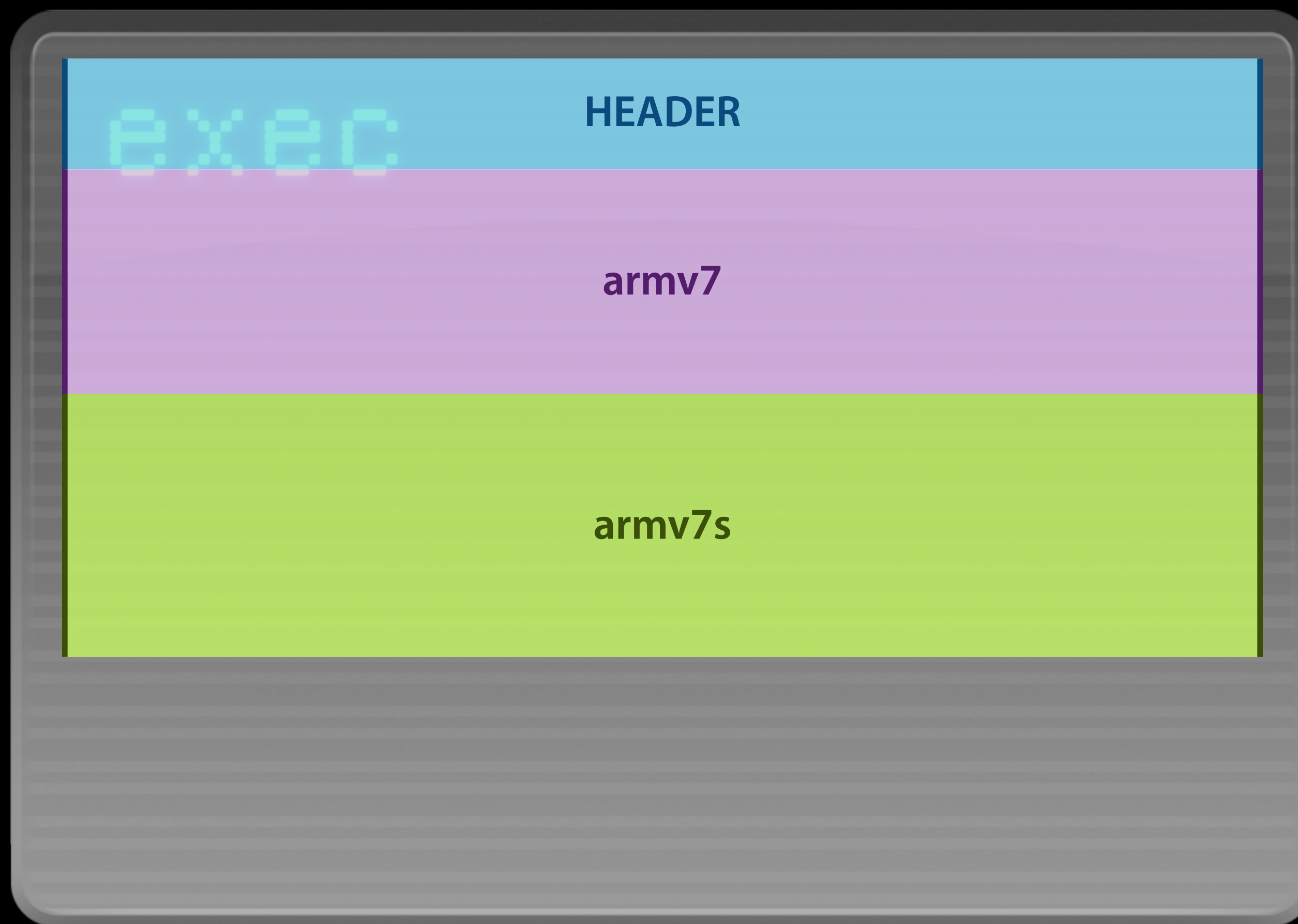
exec

HEADER

exec

HEADER

armv7







# Adopting 64-bit



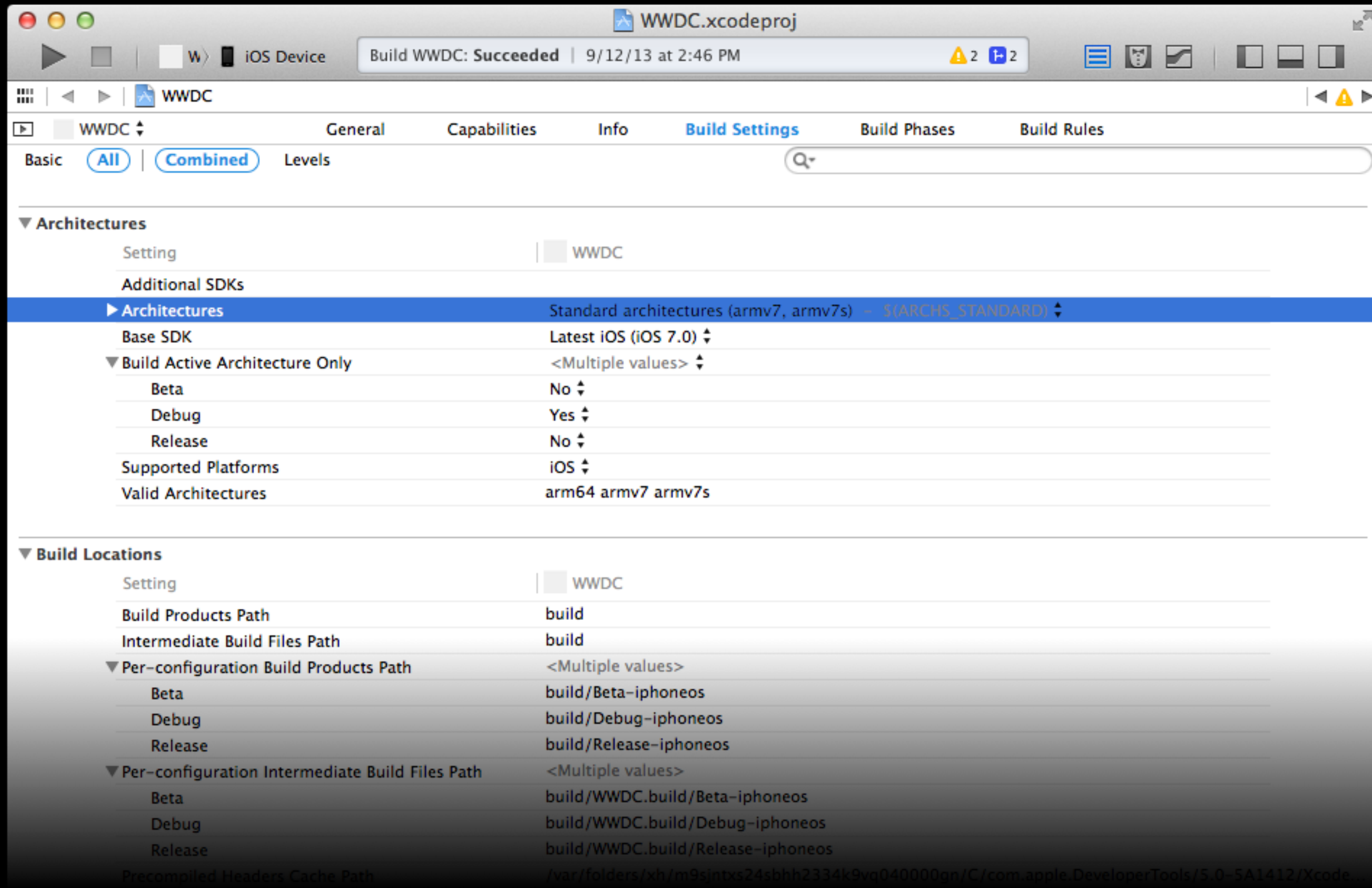
# Adopting 64-bit



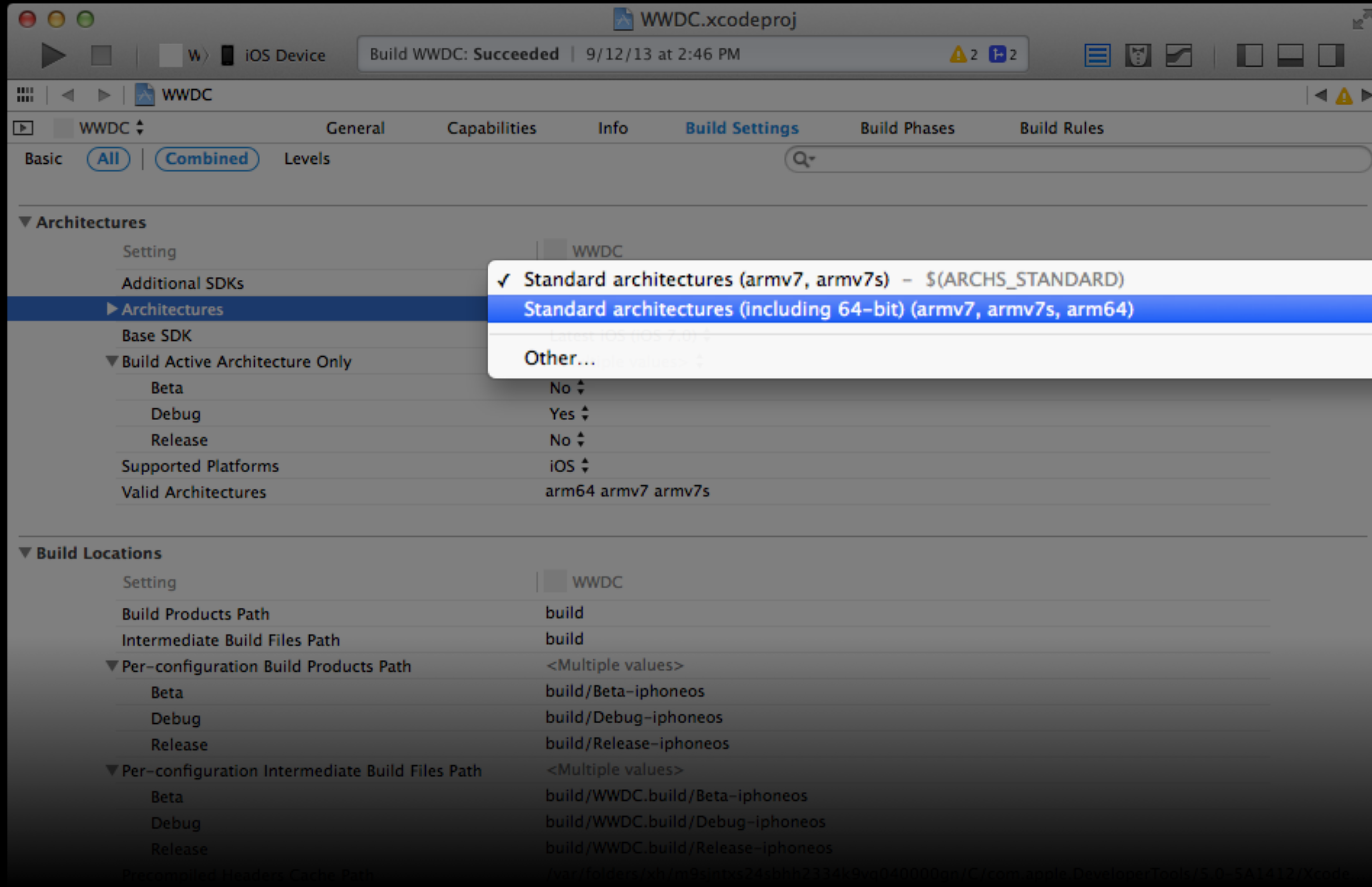
# Update Architectures



# Update Architectures



# Update Architectures



# Adopting 64-bit



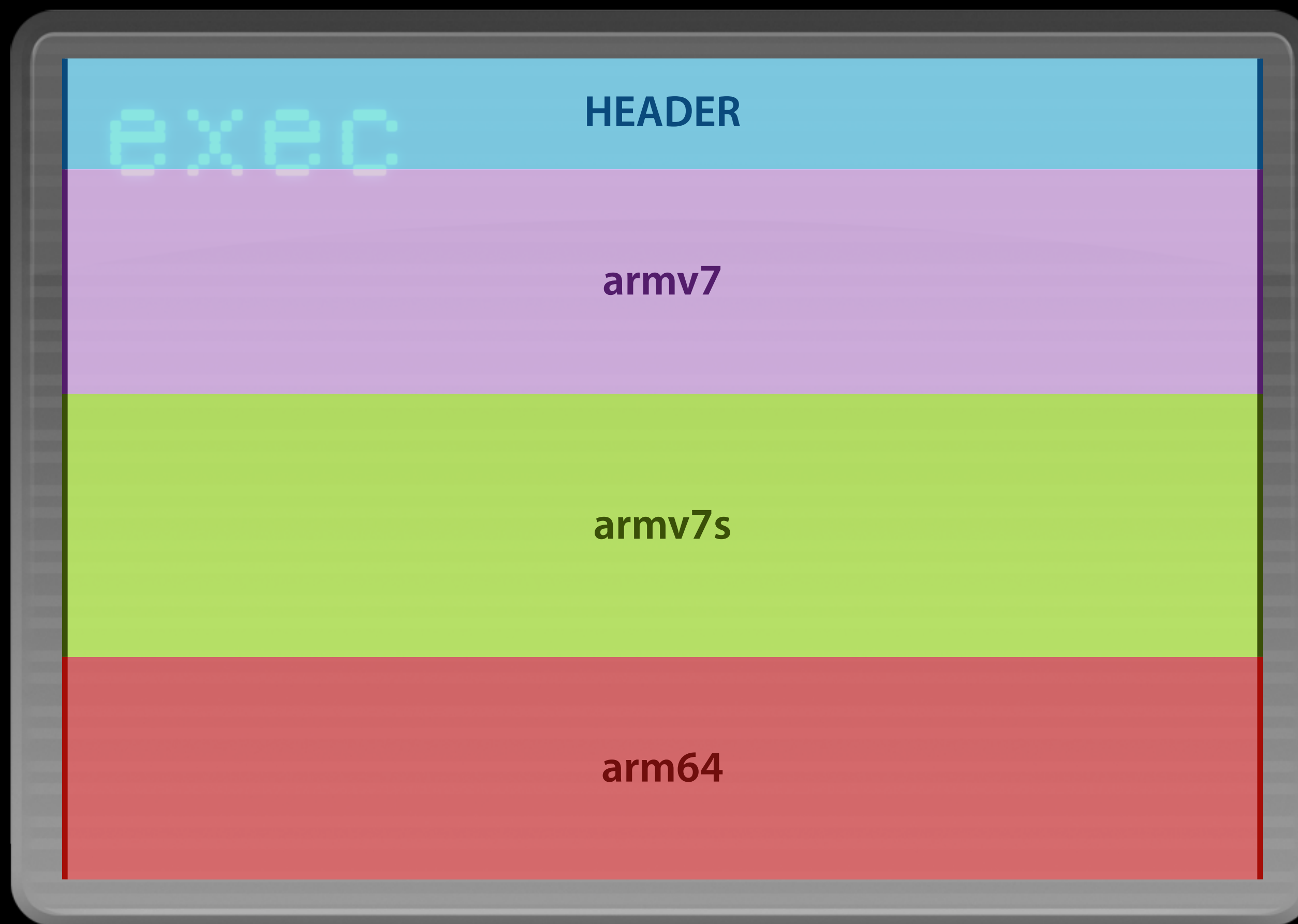
# Adopting 64-bit





# Update Libraries

- All code must be 64-bit
- Can't use 32-bit library in a 64-bit app





# Updating Libraries



# Updating Libraries

```
$file libSomeLibrary.a
```

```
libSomeLibrary.a: Mach-O universal binary with 5 architectures
```

```
libSomeLibrary.a (for architecture armv7): current ar archive random library
```

```
libSomeLibrary.a (for architecture armv7s): current ar archive random library
```

```
libSomeLibrary.a (for architecture arm64): current ar archive random library
```

```
libSomeLibrary.a (for architecture i386): current ar archive random library
```

```
libSomeLibrary.a (for architecture x86_64): current ar archive random library
```

# Updating Libraries

```
$file libSomeLibrary.a
```

```
libSomeLibrary.a: Mach-O universal binary with 5 architectures
```

```
libSomeLibrary.a (for architecture armv7): current ar archive random library
```

```
libSomeLibrary.a (for architecture armv7s): current ar archive random library
```

```
libSomeLibrary.a (for architecture arm64): current ar archive random library
```

```
libSomeLibrary.a (for architecture i386): current ar archive random library
```

```
libSomeLibrary.a (for architecture x86_64): current ar archive random library
```

# Adopting 64-bit



# Adopting 64-bit





# Primitives

# Primitives

32-bit Size (ILP32)	
<b>char</b>	1 byte
<b>BOOL</b>	1 byte
<b>short</b>	2 bytes
<b>int</b>	4 bytes
<b>long</b>	4 bytes
<b>long long</b>	8 bytes
<b>pointer</b>	4 bytes
<b>size_t</b>	4 bytes
<b>NSInteger</b>	4 bytes
<b>CGFloat</b>	4 bytes
<b>CFIndex</b>	4 bytes

# Primitives

	32-bit Size (ILP32)	64-bit Size (LP64)
<b>char</b>	1 byte	1 byte
<b>BOOL</b>	1 byte	1 byte
<b>short</b>	2 bytes	2 bytes
<b>int</b>	4 bytes	4 bytes
<b>long</b>	4 bytes	<i>8 bytes</i>
<b>long long</b>	8 bytes	8 bytes
<b>pointer</b>	4 bytes	<i>8 bytes</i>
<b>size_t</b>	4 bytes	<i>8 bytes</i>
<b>NSInteger</b>	4 bytes	<i>8 bytes</i>
<b>CGFloat</b>	4 bytes	<i>8 bytes</i>
<b>CFIndex</b>	4 bytes	<i>8 bytes</i>

# Precision

- NSInteger
  - 32-bit: -2,147,483,648 to 2,147,483,647
  - 64-bit: -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
- CGFloat
  - 32-bit: -3.4e38 to 3.4e38
  - 64-bit: -1.79e308 to 1.79e308



# 64-bit In Code

## Mismatched Types



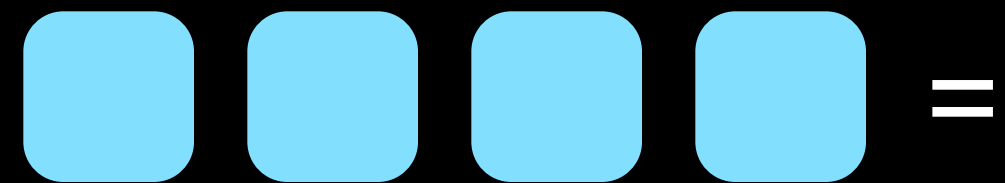
```
int myInt = [myNumber integerValue];
```

# 64-bit In Code

## Mismatched Types



```
int myInt = [myNumber integerValue];
```

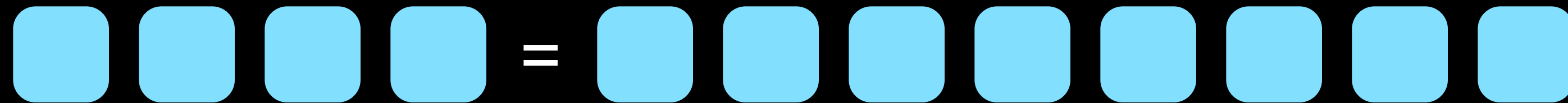


# 64-bit In Code

## Mismatched Types



```
int myInt = [myNumber integerValue];
```



# 64-bit In Code

## Mismatched Types



```
int myInt = [myNumber integerValue];
```

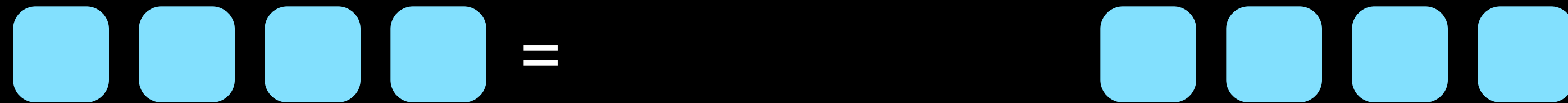


# 64-bit In Code

## Mismatched Types



```
int myInt = [myNumber integerValue];
```





# 64-bit In Code

## Mismatched Types

```
NSInteger myInteger = [myNumber integerValue];
```

# 64-bit In Code

## Mismatched Types



```
NSInteger myInteger = [myNumber integerValue];
```

# 64-bit In Code

## Mismatched Sizes



```
long myLong = 5000000000;  
memcpy(buffer, &myLong, 4);  
  
// buffer contains 4294967296
```

# 64-bit In Code

## Mismatched Sizes

```
long myLong = 5000000000;  
memcpy(buffer, &myLong, sizeof(long));  
  
// buffer contains 5000000000
```

# 64-bit In Code

## Mismatched Sizes



```
long myLong = 5000000000;  
memcpy(buffer, &myLong, sizeof(long));  
  
// buffer contains 5000000000
```

# 64-bit In Code

## Mismatched Formats



```
long myLong = 5000000000;  
NSString *s = [NSString stringWithFormat:@"%d", myLong];  
  
// s = @"705032704"
```



# 64-bit In Code

## Mismatched Formats

```
long myLong = 5000000000;  
NSString *s = [NSString stringWithFormat:@"%ld", myLong];  
  
// s = @"5000000000"
```

# 64-bit In Code

## Mismatched Formats



```
long myLong = 5000000000;  
NSString *s = [NSString stringWithFormat:@"%ld", myLong];  
  
// s = @"5000000000"
```

# Be consistent.

Use proper types, sizes, and format strings.

# 64-bit On Disk

# 64-bit On Disk

- Sharing data between devices

# 64-bit On Disk

- Sharing data between devices
  - Restoring from backup











# Read what you write.

Write 32 bits, read 32 bits.  
Write 64 bits, read 64 bits.

# 64-bit On Disk

- Sharing data between devices
  - Restoring from backup
- Read what you write
  - Write 32 bits, read 32 bits
  - Write 64 bits, read 64 bits
- Consider including metadata



# 64-bit In Memory

- Memory usage increases
  - pointers, longs, CGFloats, ...
- 32-bit only is not better


# 64-bit In Memory

- Memory usage increases
  - pointers, longs, CGFloats, ...
- 32-bit only is not better



# 64-bit In Memory

- Memory usage increases
  - pointers, longs, CGFloats, ...
- 32-bit only is not better



Operating  
System

The diagram consists of two rectangular blocks. On the left is a light blue block containing the text 'Operating System'. To its right is a much larger, dark gray block that occupies the rest of the width of the diagram. Both blocks have the same height.

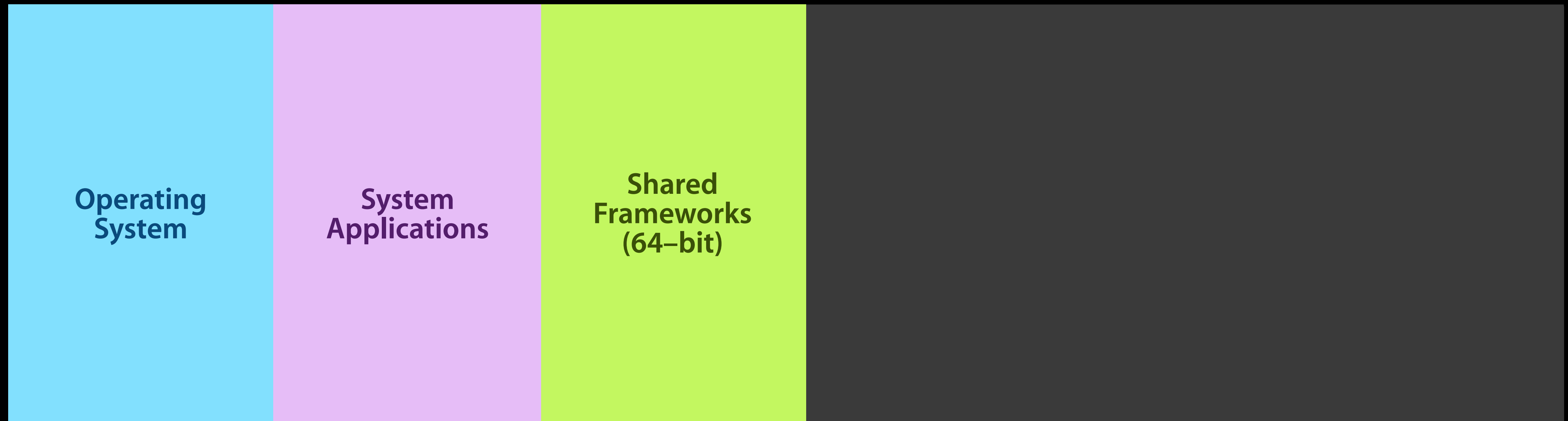
# 64-bit In Memory

- Memory usage increases
  - pointers, longs, CGFloats, ...
- 32-bit only is not better



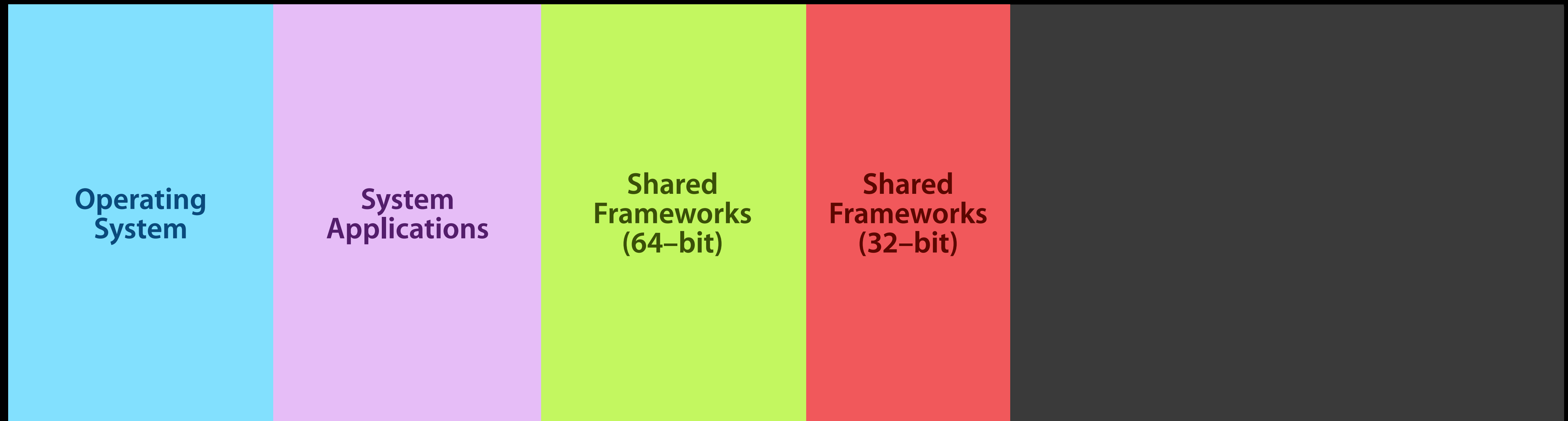
# 64-bit In Memory

- Memory usage increases
  - pointers, longs, CGFloats, ...
- 32-bit only is not better



# 64-bit In Memory

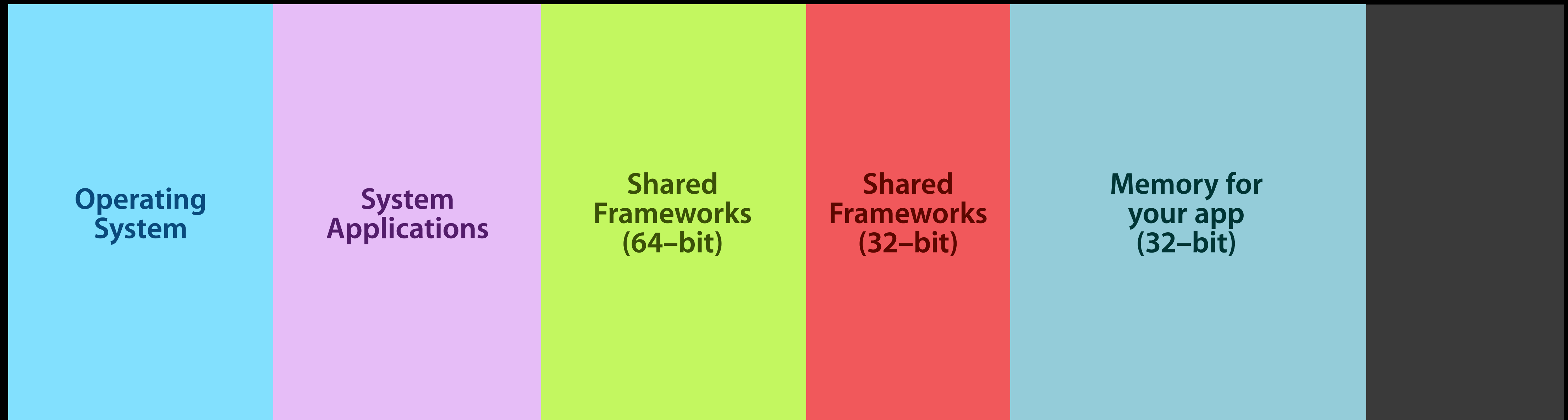
- Memory usage increases
  - pointers, longs, CGFloats, ...
- 32-bit only is not better





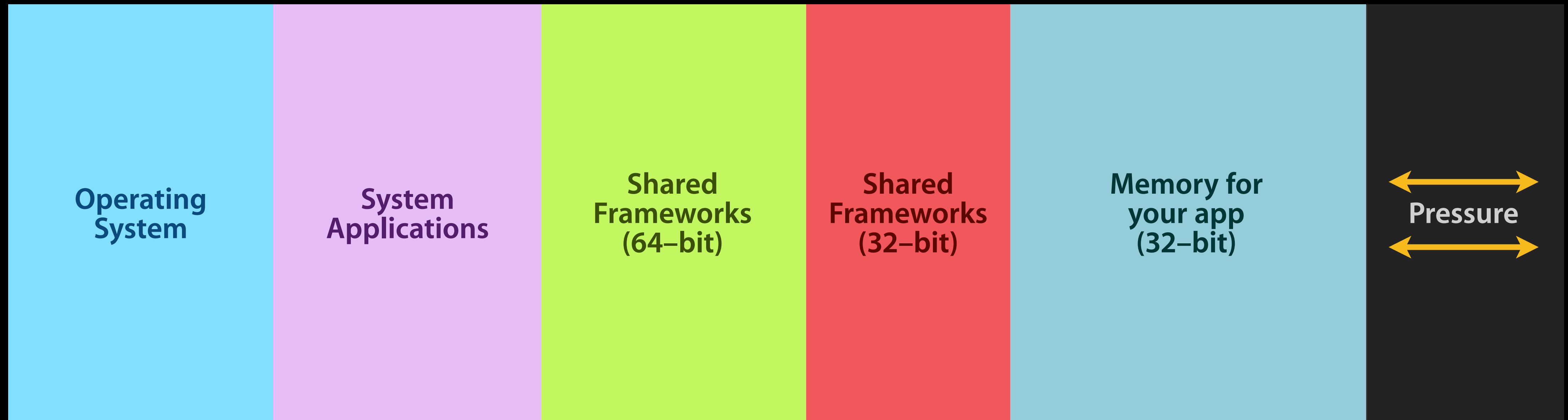
# 64-bit In Memory

- Memory usage increases
  - pointers, longs, CGFloats, ...
- 32-bit only is not better



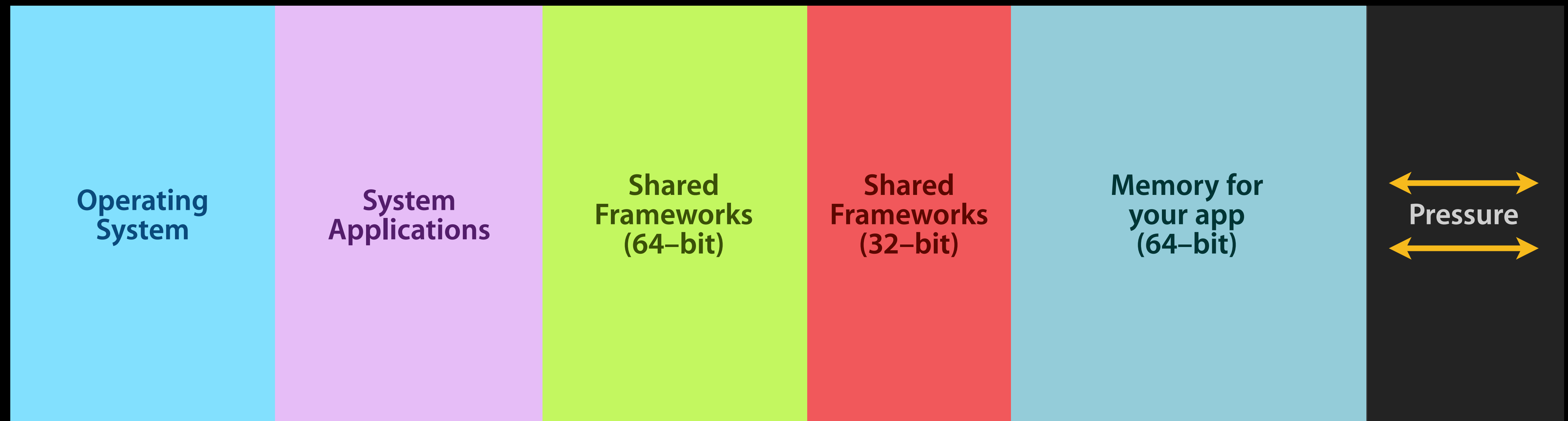
# 64-bit In Memory

- Memory usage increases
  - pointers, longs, CGFloats, ...
- 32-bit only is not better



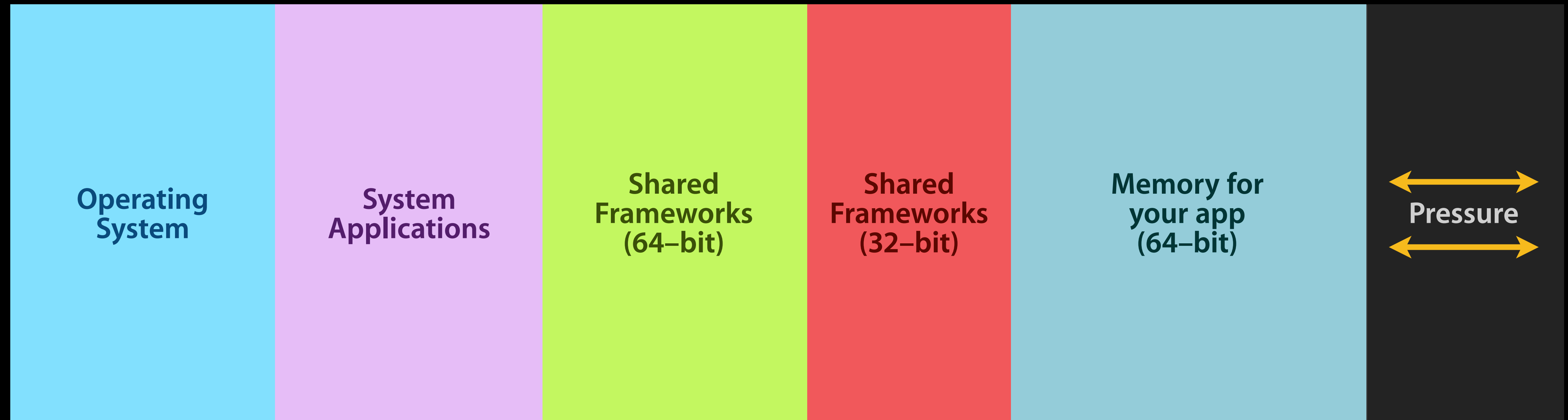
# 64-bit In Memory

- Memory usage increases
  - pointers, longs, CGFloats, ...
- 32-bit only is not better



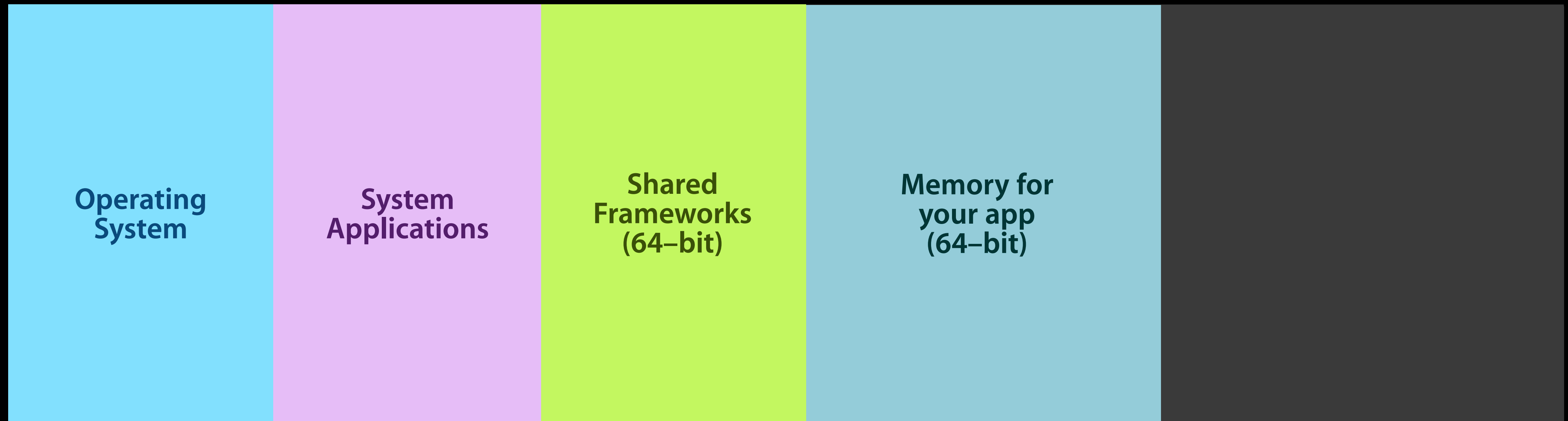
# 64-bit In Memory

- Memory usage increases
  - pointers, longs, CGFloats, ...
- 32-bit only is not better



# 64-bit In Memory

- Memory usage increases
  - pointers, longs, CGFloats, ...
- 32-bit only is not better

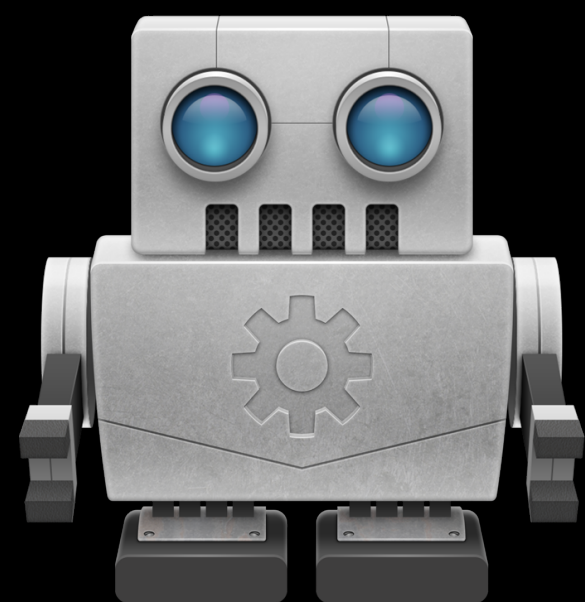


# Adopting 64-bit



# Adopting 64-bit





+



=





# 64-bit Wrap-up

- “64-bit Transition Guide for Cocoa Touch”
- Convert to 64-bit



# M7 Motion Coprocessor

- Continuously measures motion data
- Accelerometer, gyroscope, compass
- On all the time
- CoreMotion.framework

# CMMotionActivityManager

- Request periodic updates about motion
- Request motion data from a date range



```
CMMotionActivityManager *m = [CMMotionActivityManager new];

[m startActivityUpdatesToQueue:[NSOperationQueue mainQueue]
 withHandler:^(CMMotionActivity *activity) {

    // ask activity if we're running, walking, etc

}];
```



```
CMStepCounter *c = [CMStepCounter new];

[c startStepCountingUpdatesToQueue:[NSOperationQueue mainQueue]
  updateOn:1
  withHandler:^(NSInteger steps, NSError *error) {

    // the user has walked “steps” number of steps

  }];
```







# Building for iOS 7

“How do I use iOS 7 features but still support iOS 6?”

Use the iOS 7 Base SDK

Check to see if the feature exists

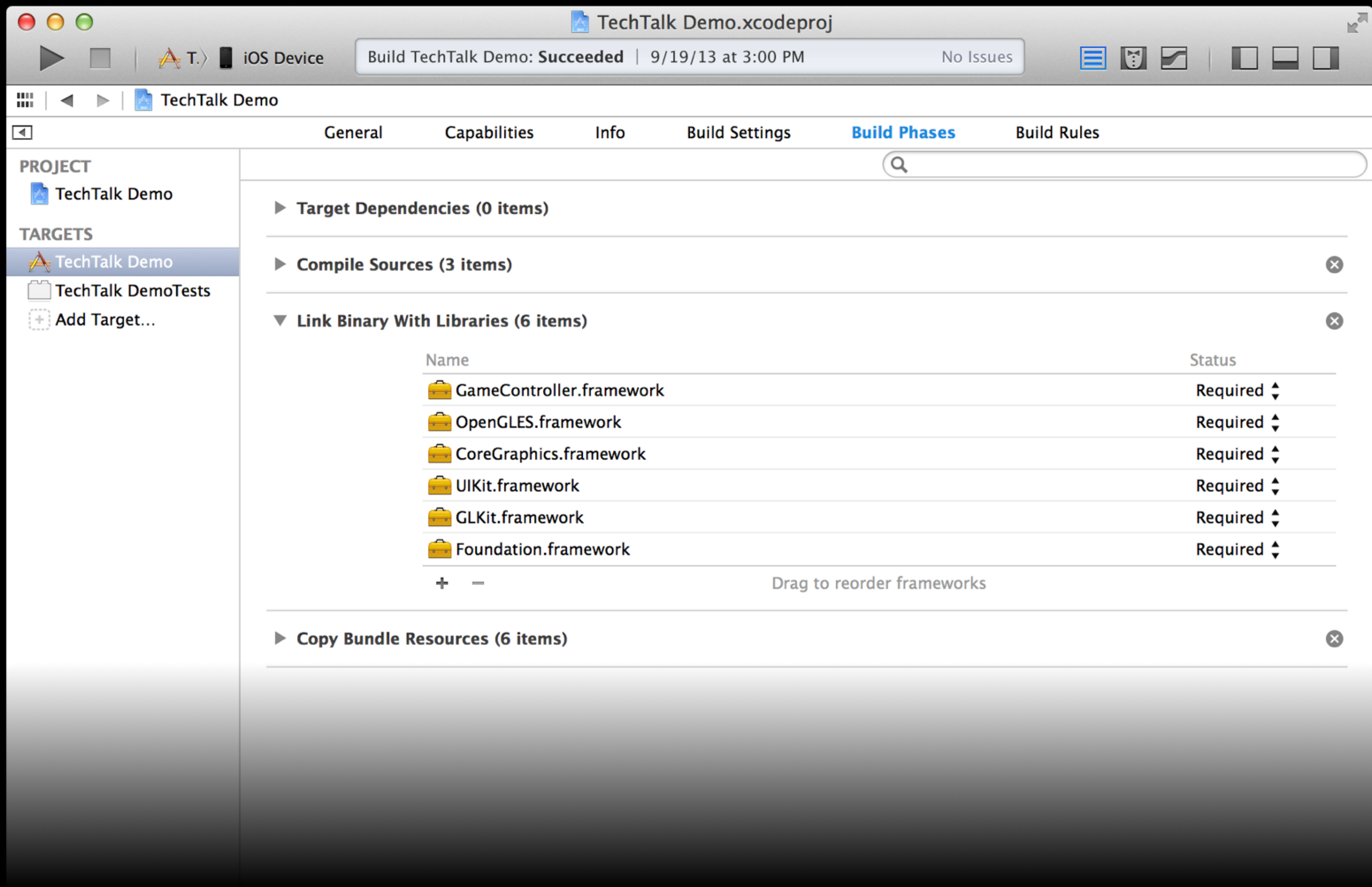
# Adopting New...

- Frameworks and classes
- Methods
- Capabilities
- Designs
- Architectures

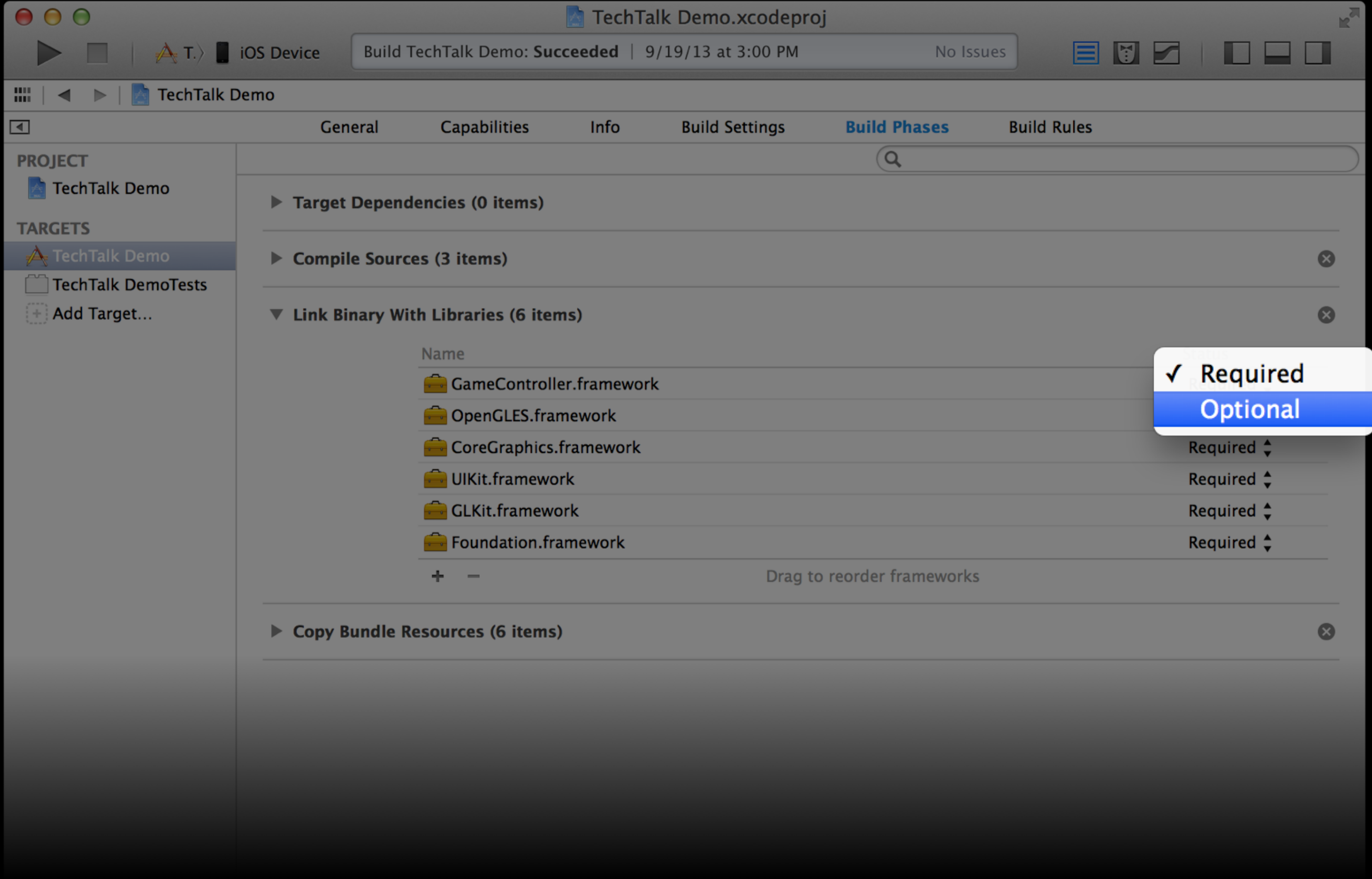
# Adopting New...

- Frameworks and classes
- Methods
- Capabilities
- Designs
- Architectures

# Optional Frameworks

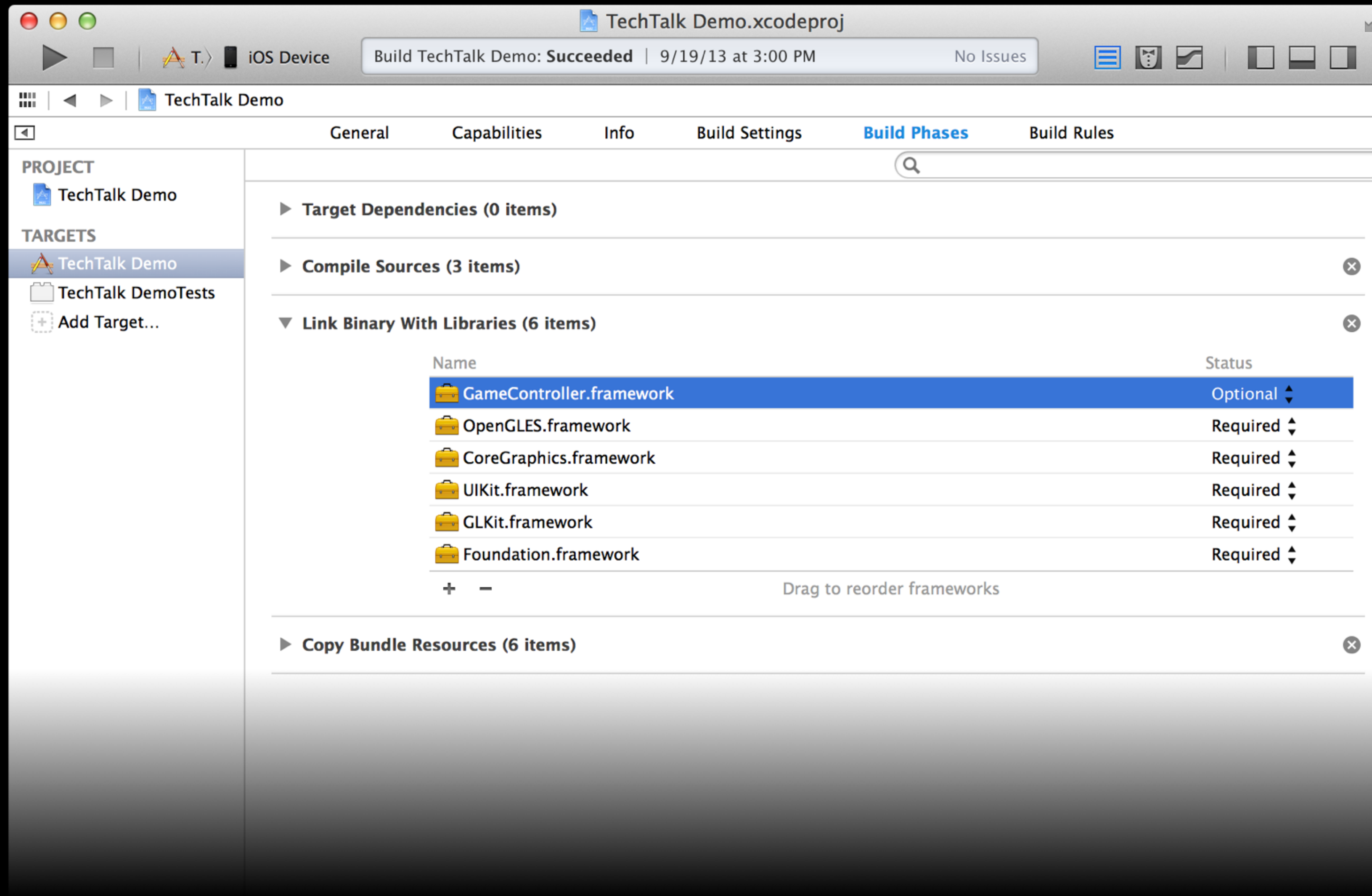


# Optional Frameworks





# Optional Frameworks





```
[GCController startWirelessControllerDiscoveryWithCompletionHandler:^(  
    // Discovery timed out or was stopped  
})];
```



```
[nil      startWirelessControllerDiscoveryWithCompletionHandler:^(  
    // Discovery timed out or was stopped  
});
```



```
[GCController startWirelessControllerDiscoveryWithCompletionHandler:^(  
    // Discovery timed out or was stopped  
})];
```



```
[GCController startWirelessControllerDiscoveryWithCompletionHandler:^(  
    // Discovery timed out or was stopped  
})];
```



```
NSNotificationCenter *center = [NSNotificationCenter defaultCenter];  
[center addObserver:self  
    selector:@selector(foundController:)  
    name:GCCControllerDidConnectNotification  
    object:nil];
```



```
NSNotificationCenter *center = [NSNotificationCenter defaultCenter];  
[center addObserver:self  
    selector:@selector(foundController:)  
    name:nil  
    object:nil];
```



```
NSNotificationCenter *center = [NSNotificationCenter defaultCenter];  
[center addObserver:self  
    selector:@selector(foundController:)  
    name:nil  
    object:nil];
```





```
if (&GCCControllerDidConnectNotification != nil) {  
    NotificationCenter *center = [NotificationCenter defaultCenter];  
    [center addObserver:self  
        selector:@selector(foundController:)  
        name:GCCControllerDidConnectNotification  
        object:nil];  
}
```



```
if (&GCCControllerDidConnectNotification != nil) {  
    NSNotificationCenter *center = [NSNotificationCenter defaultCenter];  
    [center addObserver:self  
        selector:@selector(foundController:)  
        name:GCCControllerDidConnectNotification  
        object:nil];  
}
```

# Adopting New...

- Frameworks and classes
- Methods
- Capabilities
- Designs
- Architectures



```
UIMotionEffect *effect = [self makeNewParallaxMotionEffect];  
[self.view addMotionEffect:effect];
```



```
UIMotionEffect *effect = [self makeNewParallaxMotionEffect];  
[self.view addMotionEffect:effect];
```



```
UIMotionEffect *effect = [self makeNewParallaxMotionEffect];  
[self.view addMotionEffect:effect];
```



```
SEL selector = @selector(addMotionEffect:);  
if ([self.view respondsToSelector:selector]) {  
    UIMotionEffect *effect = [self makeNewParallaxMotionEffect];  
    [self.view addMotionEffect:effect];  
}
```



```
SEL selector = @selector(addMotionEffect:);  
if ([self.view respondsToSelector:selector]) {  
    UIMotionEffect *effect = [self makeNewParallaxMotionEffect];  
    [self.view addMotionEffect:effect];  
}
```



# Adopting New...

- Frameworks and classes
- Methods
- Capabilities
- Designs
- Architectures

# Adopting New Capabilities

`+ [CMMotionActivityManager isActivityAvailable]`

`+ [MFMailComposeViewController canSendMail]`

`+ [CLLocationManager isMonitoringAvailableForClass:]`

`+ [UIPrintInteractionController isPrintingAvailable]`

`– [NSFileManager URLForUbiquityContainerIdentifier:]`

`...`

# Adopting New Capabilities

```
if ([CMMotionActivityManager isActivityAvailable]) {  
  
    CMMotionActivityManager *m = [CMMotionActivityManager new];  
    [m startActivityUpdatesToQueue:[NSOperationQueue mainQueue]  
        withHandler:^(CMMotionActivity *) {  
  
        // some motion occurred  
  
    }];  
}
```

# Adopting New...

- Frameworks and classes
- Methods
- Capabilities
- Designs
- Architectures

# Adopting New Designs

- Version check
- `NSFoundationVersionNumber`
  - `NSFoundationVersionNumber_iOS_6_1`
  - `NSFoundationVersionNumber_iOS_6_0`
  - ...
- `<Foundation/NSObjCRuntime.h>`

# Checking for Versions

```
float versionNumber = floor(NSFoundationVersionNumber);  
if (versionNumber <= NSFoundationVersionNumber_iOS_6_1) {  
    // use iOS 6-style appearance  
} else {  
    // use iOS 7-style appearance  
}
```

# Adopting New...

- Frameworks and classes
- Methods
- Capabilities
- Designs
- Architectures

# Adopting New Architectures

- Different code on different *CPUs*
- Affects compilation
- Rare to use

```
#if __LP64__  
// code that will only execute on a 64-bit device  
#else  
// code that will only execute on a 32-bit device  
#endif
```



# Hiding Feature Checks

- Don't put if-checks everywhere
- Leads to messy and unmaintainable code
- Encapsulate in wrappers
- Patterns
  - Clusters
  - Data Sources
  - Categories

# Class Clusters

- Single @interface, multiple @implementations
- Common in Cocoa: NSArray, UIButton, ...
- Private subclasses
- Encapsulate features tied to classes
- Extremely easy to remove obsolete code

# Class Clusters

# Class Clusters



TTDownloadController

# Class Clusters

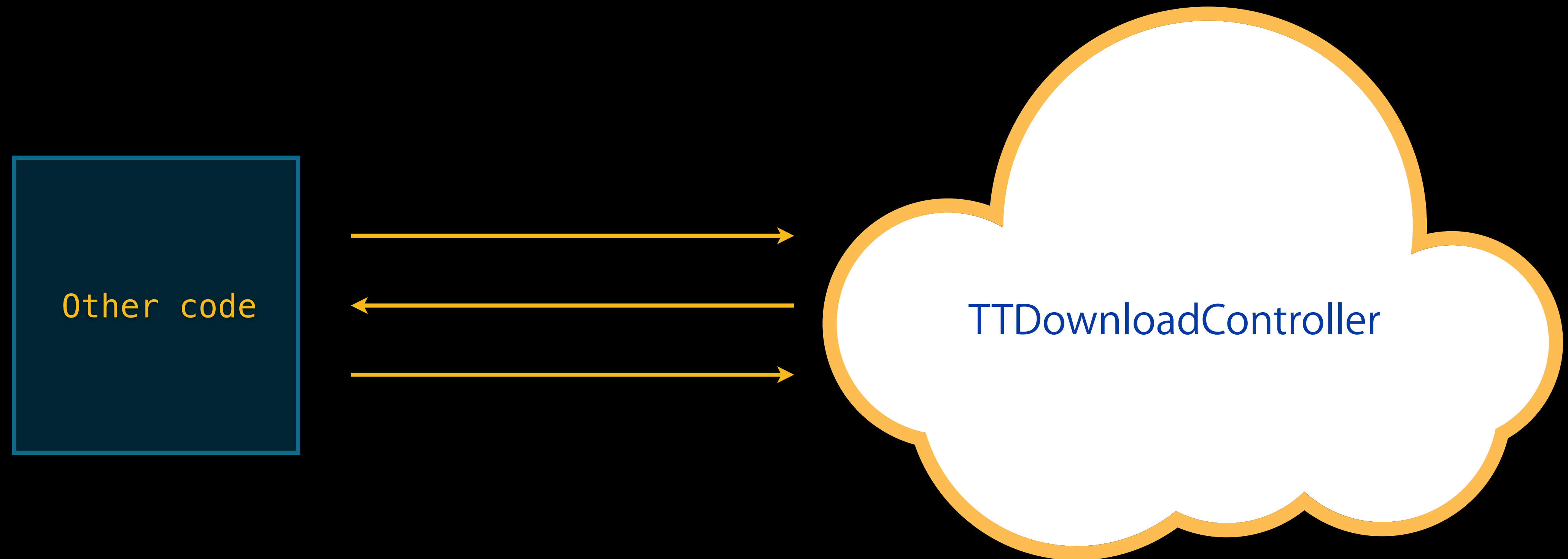


Other code

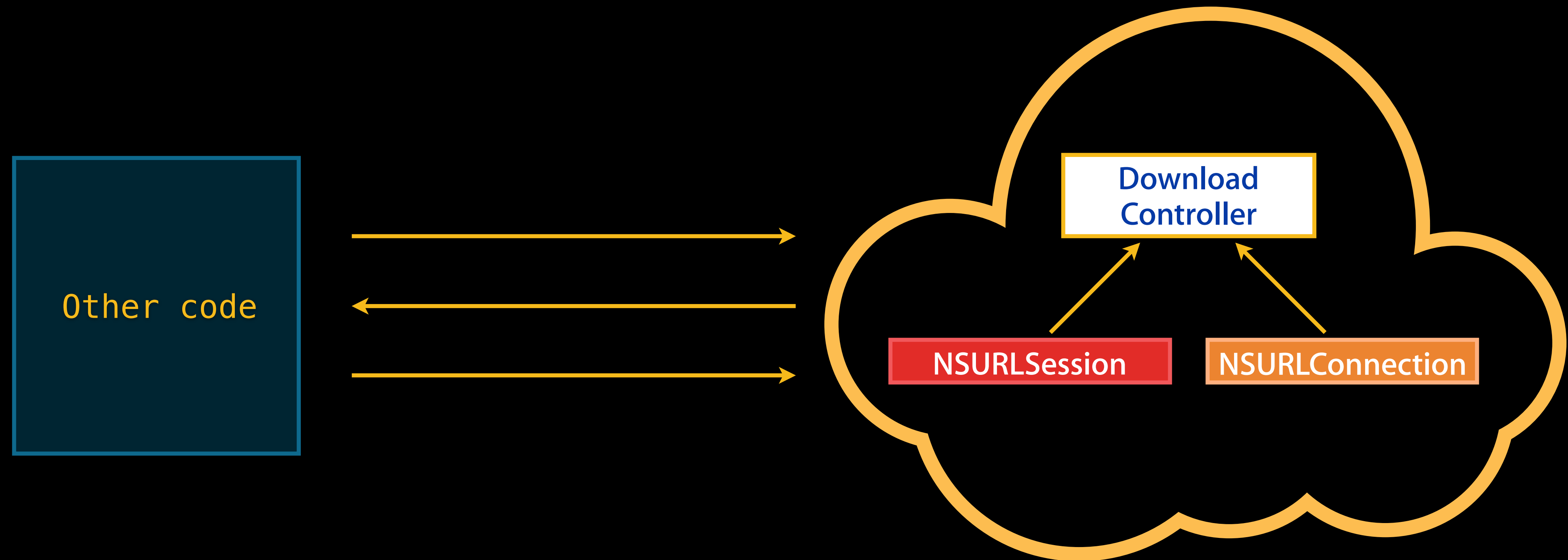


TTDownloadController

# Class Clusters



# Class Clusters



```
@interface TTDownloadController : NSObject

+ (instancetype)downloadController;

- (void)startDownload:(TTDownload *)download;
- (void)pauseDownload:(TTDownload *)download;

...

@end
```



```
@interface TTDownloadController_NSURLSession : TTDownloadController @end
@interface TTDownloadController_NSURLConnection : TTDownloadController @end

@implementation TTDownloadController

+ (instancetype)downloadController {
    if ([NSURLSession class] != nil) {
        return [[TTDownloadController_NSURLSession alloc] init];
    } else {
        return [[TTDownloadController_NSURLConnection alloc] init];
    }
}

...
@end
```

```
@interface TTDownloadController_NSURLSession : TTDownloadController @end
@interface TTDownloadController_NSURLConnection : TTDownloadController @end
```

```
@implementation TTDownloadController
```

```
+ (instancetype)downloadController {
    if ([NSURLSession class] != nil) {
        return [[TTDownloadController_NSURLSession alloc] init];
    } else {
        return [[TTDownloadController_NSURLConnection alloc] init];
    }
}
```

```
...
```

```
@end
```

```
@interface TTDownloadController_NSURLSession : TTDownloadController @end
@interface TTDownloadController_NSURLConnection : TTDownloadController @end

@implementation TTDownloadController

+ (instancetype)downloadController {
    if ([NSURLSession class] != nil) {
        return [[TTDownloadController_NSURLSession alloc] init];
    } else {
        return [[TTDownloadController_NSURLConnection alloc] init];
    }
}

...
@end
```

```
@interface TTDownloadController_NSURLSession : TTDownloadController @end
@interface TTDownloadController_NSURLConnection : TTDownloadController @end

@implementation TTDownloadController

+ (instancetype)downloadController {
    if ([NSURLSession class] != nil) {
        return [[TTDownloadController_NSURLSession alloc] init];
    } else {
        return [[TTDownloadController_NSURLConnection alloc] init];
    }
}

...
@end
```

```
@interface TTDownloadController_NSURLSession : TTDownloadController @end
```

```
@implementation TTDownloadController
```

```
+ (instancetype)downloadController {
```

```
    return [[TTDownloadController_NSURLSession alloc] init];
```

```
}
```

```
...
```

```
@end
```

# Data Sources

- Composition vs subclassing
- Data source encapsulates logic
- Encapsulate features tied to classes

```
@interface TTDownloadController : NSObject

+ (instancetype)downloadController;

- (void)startDownload:(TTDownload *)download;
- (void)pauseDownload:(TTDownload *)download;

...

@end
```

```
@interface TTDownloadController ()
@property (strong) id<TTDownloader> downloader;
@end

@implementation TTDownloadController

+ (instancetype)downloadController {
    id<TTDownloader> downloader = nil;
    if ([NSURLSession class] != nil) {
        downloader = [[TTURLSessionDownloader alloc] init];
    } else {
        downloader = [[TTURLConnectionDownloader alloc] init];
    }
    return [[self alloc] initWithDownloader:downloader];
}

...
@end
```



```
@interface TTDownloadController ()
```

```
@property (strong) id<TTDownloader> downloader;
```

```
@end
```

```
@implementation TTDownloadController
```

```
+ (instancetype)downloadController {
```

```
    id<TTDownloader> downloader = nil;
```

```
    if ([NSURLSession class] != nil) {
```

```
        downloader = [[TTURLSessionDownloader alloc] init];
```

```
    } else {
```

```
        downloader = [[TTURLConnectionDownloader alloc] init];
```

```
    }
```

```
    return [[self alloc] initWithDownloader:downloader];
```

```
}
```

```
...
```

```
@end
```

```
@interface TTDownloadController ()
@property (strong) id<TTDownloader> downloader;
@end

@implementation TTDownloadController

+ (instancetype)downloadController {
    id<TTDownloader> downloader = nil;
    if ([NSURLSession class] != nil) {
        downloader = [[TTURLSessionDownloader alloc] init];
    } else {
        downloader = [[TTURLConnectionDownloader alloc] init];
    }
    return [[self alloc] initWithDownloader:downloader];
}

...
@end
```

```
@interface TTDownloadController ()
@property (strong) id<TTDownloader> downloader;
@end

@implementation TTDownloadController

+ (instancetype)downloadController {
    id<TTDownloader> downloader = nil;
    if ([NSURLSession class] != nil) {
        downloader = [[TTURLSessionDownloader alloc] init];
    } else {
        downloader = [[TTURLConnectionDownloader alloc] init];
    }
    return [[self alloc] initWithDownloader:downloader];
}

...
@end
```

```
@interface TTDownloadController ()
@property (strong) id<TTDownloader> downloader;
@end

@implementation TTDownloadController

+ (instancetype)downloadController {
    id<TTDownloader> downloader = nil;

    downloader = [[TTURLSessionDownloader alloc] init];

    return [[self alloc] initWithDownloader:downloader];
}

...
@end
```

# Categories

- Add methods to framework classes
- Encapsulate features tied to methods



```
UIMotionEffect *parallaxEffect = [self makeNewParallaxMotionEffect];  
  
[self.view addMotionEffect:parallaxEffect];
```

```
@interface UIView (TechTalkCompatibility)

- (void)tt_addMotionEffect:(UIMotionEffect *)effect;

@end
```

```
@implementation UIView (TechTalkCompatibility)
```

```
- (void)tt_addMotionEffect:(UIMotionEffect *)effect {  
    if ([self respondsToSelector:@selector(addMotionEffect:)]) {  
        [self addMotionEffect:effect];  
    }  
}
```

```
@end
```

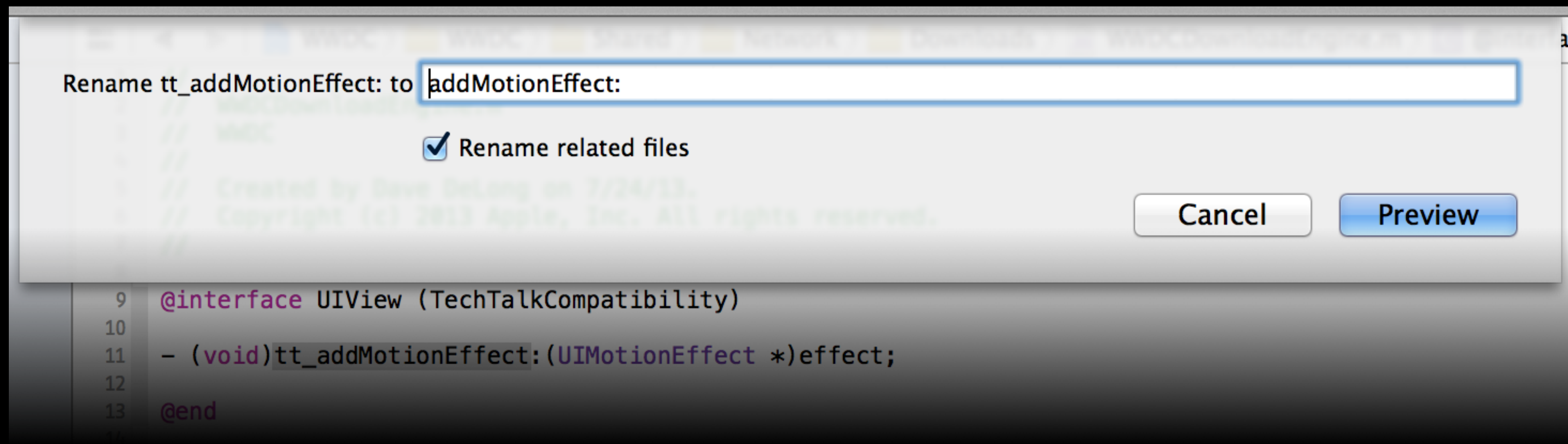


```
UIMotionEffect *parallaxEffect = [self makeNewParallaxMotionEffect];
```

```
[self.view tt_addMotionEffect:parallaxEffect];
```

# Categories

- Add methods to framework classes
- Hide features tied to methods
- Remove via Edit ► Refactor ► Rename...



# Backporting Features

- Some features can be approximated
- Some are very hard to duplicate
- Scale back experience on older OSes



